ACCREDITATION SUPPORT PACKAGE

FOR

JOINT SERVICES ENDGAME MODEL (JSEM)

ASP - I – MODEL STATUS REPORT

Joint Accreditation Support Activity (JASA)
Naval Air Warfare Center, Weapons Division (418100D)
China Lake, CA

JTCG/AS-96-M-00 2

FOREWORD

This document was developed by the Joint Accreditation Support Activity (JAS.) technical support office at the Naval Air Warfare Center, Weapons Division (NAWCWPNS), under the auspices of the Joint Technical Coordinating Group on Aircraft Survivability (JTCG/AS). The effort reported herein was funded by the Requirements Directorate of the Joint Strike Fighter (JSF) program as a part of their on-going Model and Simulation accreditation efforts .

Executive Summary

This is an initial version of the Accreditation Support Package Volume I (ASP-I) for the Joint Services Endgame Model (JSEM). It is designed to provide a potential user with a characterization of the current state of JSEM with respect to criteria related to its general acceptability for use. The information collected here should characterize the model well enough to provide an initial determination of its suitability for particular applications. It should also provide confidence that the model is well enough managed and supported to yield consistent results across its spectrum of users and applications.

The information provided to characterize JSEM consists of the following elements:

a. A description of the configuration management baseline for the model. JSEM is well-managed by a tri-service group under the JTCG/AS and JTCG/ME, which controls changes to a baseline version distributed through JTCG/ME. The model is distributed under the control of the JTCG/ME, and identical versions embedded in other survivability analysis tools are distributed through SURVIAC.

b. A summary of implicit and explicit assumptions and limitations inherent in the model. A listing of known errors or anomalies found as a result of prior V&V efforts is also included. JSEM has some known limitations which may affect its suitability for use, primarily in its associated proximity fuze simulations.

c. A review of the model's development, verification and validation (V&V) and usage history. JSEM has a wide user base, but very little documented V&V. It is recognized as the standard tri-service missile endgame analysis tool.

d. A review of the status of model documentation and its conformity to accepted software documentation standards. JSEM User Manuals are good and meet the intent of applicable standards, but the analyst manual is a draft version only, and there is no software design documentation available (no conceptual model specification). Documentation is one of the recognized deficiencies of JSEM. Even with the help of the JSEM pre-processor software (JPREP), it is advisable for any new user to seek the assistance of an experienced JSEM user before applying the model to a problem.

e. A summary of overall software quality as characterized by conformance to accepted design and coding practices. Generally, JSEM code is modular, with some significant exceptions. The general consensus is that JSEM software quality rates a grade of "B", even though it was "written by committee" and could be modernized.

In summary, this ASP-I provides the details of these information elements in a single document. The degree to which each information element is complete and current provides a general indication of whether JSEM is suitable for further consideration for use in a particular application. However, it should be noted that this document was compiled over a very short period of time and with very limited resources. As a consequence, there may be additional relevant data in the community which would shed more light on the status of JSEM. It is recommended that this document be updated as those data and more resources become available.

TABLE OF CONTENTS

1.0 Phase I Accreditation Support Package Description

This document describes the results of five distinct activities: definition of JSEM's configuration management baseline; summarization of model assumptions, limitations and errors; determination of the model's V&V status and usage history; assessment of available documentation; and assessment of software quality. The information elements contained in an ASP I were defined by an extensive survey and analysis of tri-service and industry M&S user requirements for information to support an accreditation decision[1] [2].

2.0 Configuration Management (CM) Baseline

The configuration management (CM) baseline description for a model provides prospective users with an indication of how well the model is controlled and supported. Models with poorly defined configurations and unspecified (or vague) change control procedures are likely to produce inconsistent results across the spectrum of users and applications, with the consequence that model predictions will not be highly regarded. Models whose configurations are well specified, and whose change procedures are well disciplined are more likely to have timely supporting documentation and to produce consistent, well accepted results. Moreover, well managed models have a lower risk of failing detailed V&V aimed at higher levels of accreditation.

The CM baseline for a model consists of a description of the model, its development history, current version status (including documentation), applicable change procedures, model development policy (including beta site version integration), and any configuration management policies, procedures, guidelines and support functions in place for the model. Taken as a whole, these information elements provide the prospective user with a vantage point from which to assess the discipline with which a model has been developed, the important operational differences between extant versions, and the potential impact of model management discipline on the acceptability of model results.

The following sections describe the JSEM model, a little about its development history, and summarizes how JSEM is managed.

2.1 Model Description.

The Joint Service Endgame Model (JSEM) is the standard Joint Technical Coordinating Group for Munitions Effectiveness (JTCG/ME) and the JTCG on Aircraft Survivability (JTCG/AS) missile endgame (fuze and warhead) simulation for use against air targets. JSEM is used to supply endgame probability of kill (Pk) or hit (Ph) information for threat systems against US air vehicles and for US missiles against foreign air targets. It is used to generate all the endgame Pk data for use in anti-air Joint Munitions Effectiveness Manuals (JMEMs).

---

[1] Joint Technical Coordinating Group for Aircraft Survivability (JTCG/AS). *A Comparative Analysis of Tri-Service Accreditation Policies and Practices: Volume I of the Accreditation Requirements Study Report*, by Dennis Laack. Computer Sciences Corp, Camarillo, CA, for the SMART Project, NAWCWPNS, China Lake, CA, February 1994. 39 pp. + appendices (JTCG/AS-93-SM-020, publication UNCLASSIFIED.)

[2] Joint Technical Coordinating Group for Aircraft Survivability (JTCG/AS). *Information Requirements in Support of Accreditation: Volume II of the Accreditation Requirements Study Report*, by Dennis Laack. Computer Sciences Corp, Camarillo, CA, for the SMART Project, NAWCWPNS, China Lake, CA, February 1994. 27 pp. + appedices (JTCG/AS-93-SM-020, publication UNCLASSIFIED.)

JSEM simulates the missile fuze and warhead functions, and make use of off-line generated vulnerability models of the aircraft to generate a Pk and/or a Ph estimate. The endgame is the very last portion of the missile's flight as the missile approaches its air target, representing approximately one guidance time constant of flight prior to intercept. The fuze sensor interaction with the air vehicle target is modeled, the burst control logic is exercised to determine a detonation point, the warhead fragments are "flown" out to the target, and blast, fragment and structural kill effects on the aircraft are evaluated. The warhead characterizations are based, where possible, on test results for actual detonated warheads; where the actual warhead characterization has not been measured, estimates of warhead characteristics are obtained either from the warhead designer (for US systems), or (for foreign warheads) those estimates are based upon the level of technology available at the time the system was developed.

Warhead types modeled include: blast-fragmentation, naturally fragmenting, controlled fragmentation, discrete rods, azimuthally aimable versions of those, and continuous rods.

Existing fuze simulations used with JSEM can be categorized [for the most part] into three groups: simple geometric approximations (available internal to JSEM itself), nearfield prediction algorithms, and empirically based models for very specific fuzes and targets. Empirical models can be very effective depending upon the care with which they are constructed. They are for specific systems and therefore not necessarily of general interest. Lethality (Blue) fuze simulations are usually based on measurements and are generally very good at predicting fuzing for a specific [Blue] system.

Simple geometric codes rely on simple cone-like representations of the fuze sensory pattern intersecting a geometric target or predetermined glitter points on a target. These models work well enough in some cases. However, there are several problems associated with these approaches. Perhaps the biggest problem is that within some predetermined fuze cut-off range the probability of fuzing P(F) [as assumed by the stick-cone model] is always one. For physically large and complex targets that assumption may not be critical. However, the assumption that P(F) = 1 for a "small" target may be a fatal simulation error. The fuzing factor [in the survivability equation] can vary from zero to one for small targets and can be the driving element in the P(K) calculation.

For the past several years two nearfield RCS prediction algorithms have been available for use with JSEM, by running them "off-line" and importing fuze point files into JSEM. The Geometric Theory of Diffraction (GTD) approach is particularly appealing because of its relatively fast run times. It is, however, limited to small targets because the target models must be built up from primitive geometric shapes (spheres, plates, cones, etc.). Physical Theory of Diffraction (PTD) target models can be generated from complex shapes using CAD target descriptions. However, PTD has prohibitively long computer run times at this stage of its application to the fuze problem. GTD assumes that all the reflected and diffracted energy is coming from one (easily calculated) geometric location on each target element. This simplifying assumption is the basis for the fast GTD runtimes. PTD uses an approach that integrates the energy from many small grid cells representing the target surface. The interaction with large numbers of surface cells causes long computer run times.

The vulnerability estimates used in JSEM usually are based upon the COVART model. COVART (Computation of Vulnerable Areas and Repair Times) calculates vulnerable areas for each critical component in the aircraft for single penetrators (projectiles and warhead fragments). For each critical component in the target, vulnerable area tables are input , along with a "fault tree" which describes the effect of damaging each component on the ability of the aircraft to survive. These vulnerable area tables have been developed for a number of aircraft over the years, both for foreign and domestic aircraft.

JSEM has been successfully run by a variety of users on VAX, SGI, HP, SUN, Data General and CDC workstations. It has also been run on a CRAY, and there is a PC Version available. Most

current users have UNIX workstations. The memory requirement for running JSEM is a function of how large the arrays are made to support vulnerability inputs; memory is not generally a problem.

2.2 Development History.

JSEM development began in the late 1980's as a follow-on to earlier work in endgame model standardization sponsored by the JTCG/ME Missile Evaluation Group (MEG). Back in 1973, the MEG sponsored a review of a number of endgame models in use at that time (Figure 1), with the objective of developing a new, standard "reference model." The result of that effort around 1979 was the REFMOD simulation, used for a time to develop Pk estimates for JMEM documents. A number of shortfalls of the REFMOD simulation (notably user unfriendliness) led to a further development, called Modular Endgame Computer Algorithms (MECA). MECA, completed in 1983, was primarily an effort to modularize the REFMOD software and make it more user friendly. The success of this effort at building a standardized endgame tool can be measured by the fact that the Air Force at Eglin AFB decided to concurrently develop the SHAZAM endgame model.

The primary difference betwen SHAZAM and MECA was that SHAZAM modeled the warhead fragmentation by individually tracing every fragment path from the warhead to the target, while MECA calculated the expected number of fragment hits on each component (using polar and radial zones to describe the warhead fragmentation pattern).

Fuzing routines for MECA were all maintained external to the MECA software, while SHAZAM required internal fuzing routines. Fuzing for SHAZAM was provided primarily by a model development called "Backscatter Electronic Event Simulation Technique (BEEST)" for radio frequency (RF) fuzes. An Active Optical Target Detector (AOTD) model was also available. For MECA, external routines were available based on the Geometrical Theory of Diffraction (GTD) for RF fuzes, an empirically derived AOTD model, and the simple geometric representation of the fuze pattern that is available in JSEM today.

Because of the concurrent developments of MECA and SHAZAM, the JTCG/ME funded the development of the JSEM model, which is derived from both of them. JSEM contains all the functionality of both models under a single architecture. The most recent development is the entry of JSEM into the Survivability/Vulnerability Information Analysis Center (SURVIAC) model repository as the accepted standard missile endgame model by JTCG/ME and JTCG/AS.

Figure (2) illustrates the development of JSEM in more detail from the initial development of MECA through the present version of JSEM. As the figure suggests, JSEM is provided with a pre-processor (JPREP) which makes it easier to set up input data files and ensure proper execution of the code.

2.3 Version Description and Current Status.

The current version of JSEM is version 2.2, which is the version referenced if a user contacts SURVIAC for the model. Appendix A contains a summary of the changes from version 2.1 to version 2.2. The changes included bug fixes, improved missile debris representation, updates to fuzing, encounter and contact models, and the addition of elements required to use JSEM to generate Pk's for ship point defense gun systems. Appendix B contains those changes from version 2.2 to version 2.2.2, which will become the next baseline version during FY97. The

configuration items include the source code, the JPREP pre-processor, and volumes I and II of the users manual[3]. These are not tracked by configuration item numbers.

JSEM development is conducted through the Oklahoma State University Eglin Field Office, and distributed through Tinker AFB, under the auspices of the JTCG/ME[4]. Along with the simulation, a number of standard sample test cases are provided on the diskette. The code itself is unclassified, as are the sample cases. Access to the code is restricted to those with a need to know, and a disclaimer is required to be on file with the OSU Field Office before the code can be distributed. FORTRAN source code is provided to the user. The form for the disclaimer statement is shown in Figure 3. JSEM has also been integrated into the ESAMS 2.8 surface-to-air missile model and into the Digital Integrated Modeling Environment (DIME); it is distributed with both those codes through SURVIAC.

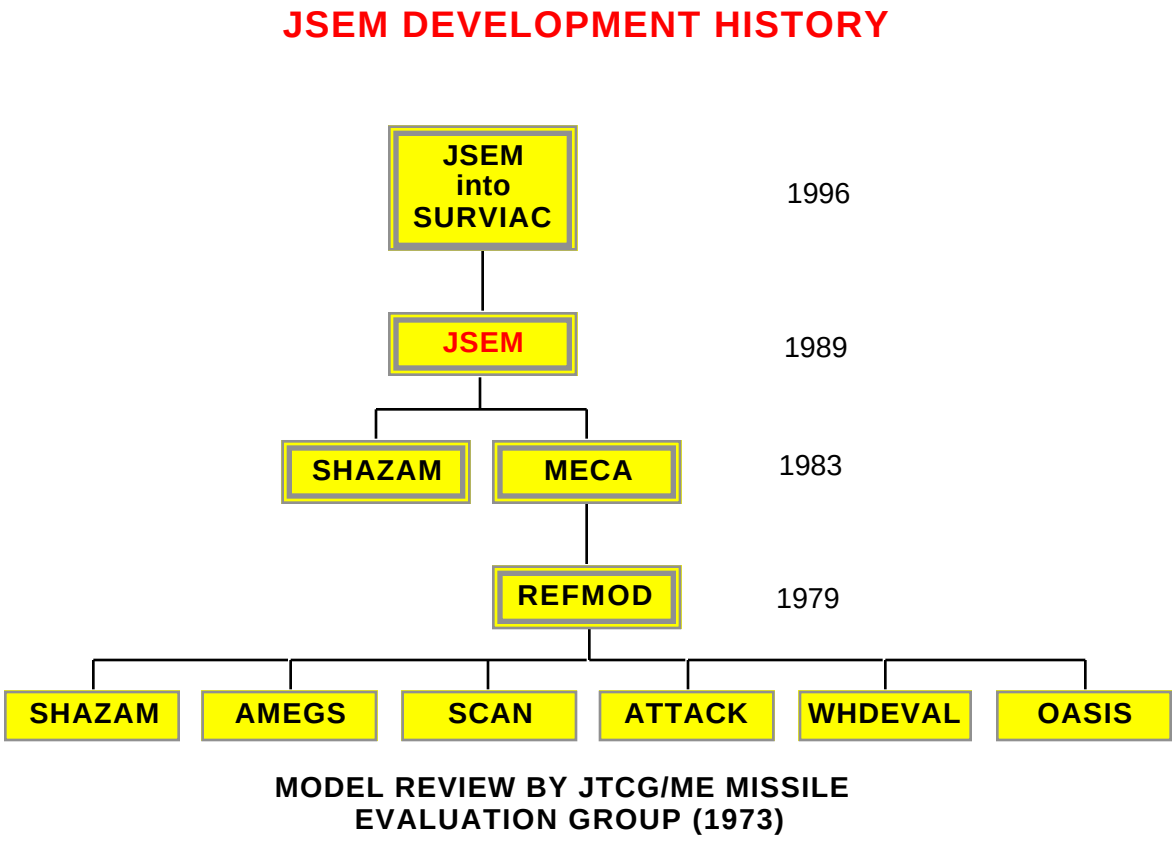

Figure 1 JSEM Development History

[3] Joint Services Endgame Model (JSEM) Computer Program, Volume I - An Introduction to JSEM (Getting Started), 94 pp., 61JTCG/ME-88-3-1, 30 June 95.
Joint Services Endgame Model (JSEM) Computer Program, Volume II - Reference Manual, 396 pp., 61JTCG/ME-88-3-2, 30 June 95.

[4] Joint Services Endgame Model (JSEM) Computer Program, Version 2.2 (Magnetic Diskette), 61JTCG/ME-88-3 (source code and sample cases) 2 August 95

2.4 Change Procedures.

By direction from the JSEM Configuration Control Board (CCB) Chairman, the JSEM code is maintained by the Eglin Field Office of Oklahoma State University (OSU). The duties of OSU are to:

a. Archive old versions of the code
b. Incorporate changes and modifications as suggested by users and approved by the CCB
c. Write and edit User and Analyst Manuals
d. Maintain sample test cases for ensuring proper code operation
e. Perform studies using JSEM, as directed (and funded) by JTCG/ME

The CCB, which is composed of the individuals listed in Table 1, meets at least once a year. The functions of the CCB are:

a. Discuss and vote on all change items submitted to OSU for consideration
b. Approve changes to be incorporated into next version of the code

There is a User Group for JSEM, composed of approximately 50 users, which also meets at least once a year. During the user group meeting, different users brief the group on their use of the code, and any problems or needed improvements that they have identified. The User Group submits suggestions for improvement or code corrections to the CCB for consideration. The User Group includes personnel from the following DoD organizations:

Army Materiel Systems Analysis Activity (AMSAA)
Army Research Laboratory (ARL)
Air Force Aeronautical Systems Center (ASC)
Naval Surface Warfare Center (NSWC)
Naval Air Warfare Center, Weapons Division (NAWCWPNS)
US Army ARDEC
Wright Laboratory (WL/FIVS)
57th Test Group TGOA
US STRATCOM
Naval Warfare Assessment Division
ALC TISFA, Hill AFB
Air Force Studies and Analysis Agency (AFSAA)
US Army Missile Command

Industry , Academia and other government representation on the User Group is from the following organizations:

Alliant Techsystems Inc
ASI Systems International
Boeing Military Airplane
Institute for Defense Anaysis (IDA)
LORAL Vought Systems
LORAL Aeroneutronics
Motorola, Inc
PMC, Inc
Raytheon
SURVICE Engineering
Oklahoma State University
Texas Instruments
Northrop Grumman Corporation
Booz•Allen and Hamilton

Naval Postgraduate School
McDonnell Douglas Aerospace Company
ENTEK Inc
BDM Federal
Ketron
Lockheed
Simulation Technologies Inc
KBM Enterprises, Inc
Central Intelligence Agency
Sandia National Laboratory
Orlando Technology, Inc

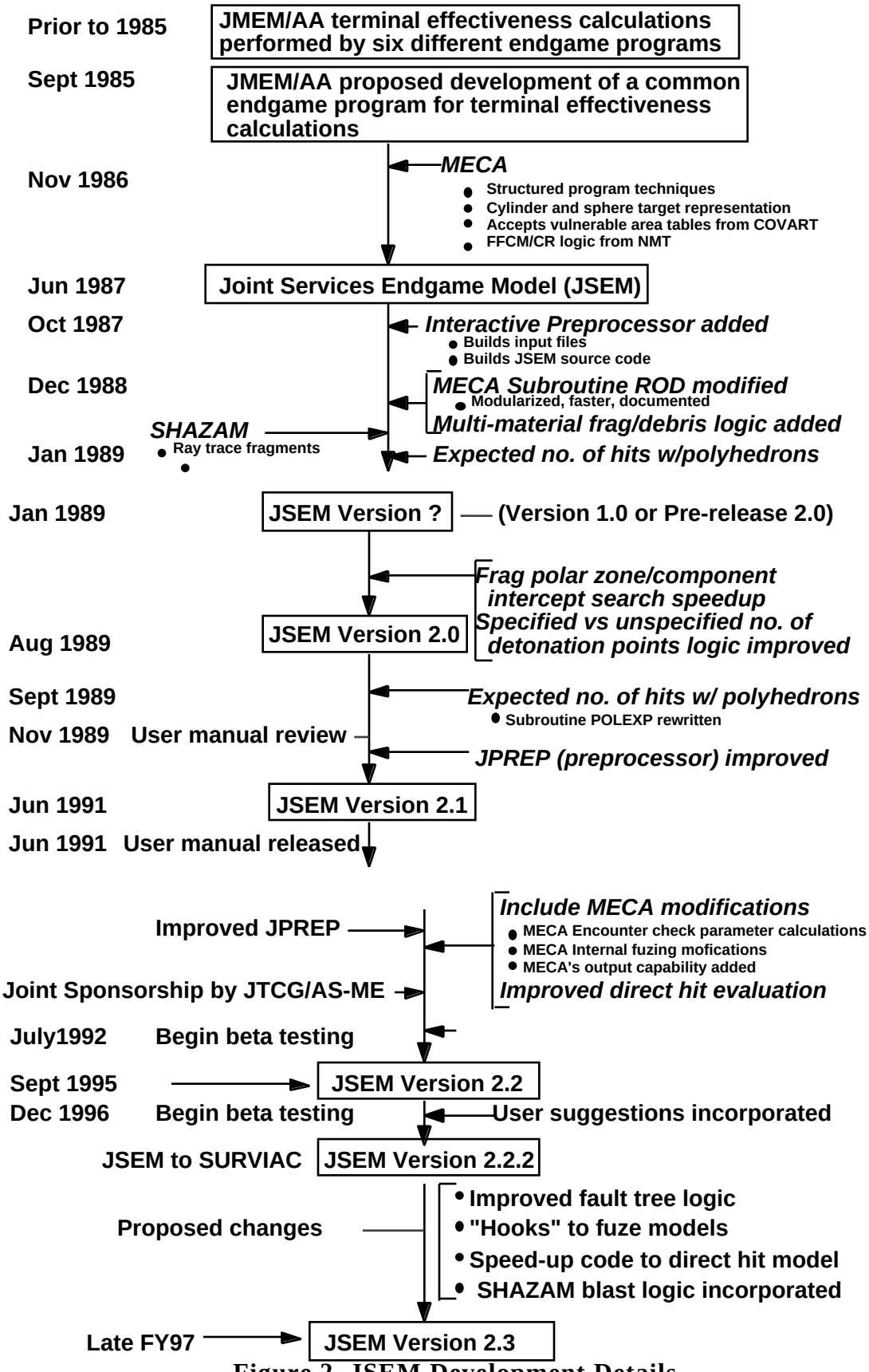

Figure 2 JSEM Development Details

Joint Technical Coordinating Group for Munitions Effectiveness
(JTCG/ME)

STATEMENT OF TERMS AND CONDITIONS FOR RELEASE OF GOVERNMENT
OWNED OR DEVELOPED COMPUTER SOFTWARE

1. I/We request a copy of the United States Government owned software package, Joint Services Endgame Model (JSEM).

2. The requested software package will be used for the following purposes: (Non-Government: Include one or more current DOD contract numbers and expiration dates for which the computer program is required.)

Such use is projected to accrue benefit to the Government as follows:

3. I/we will be responsible for assuring that the software we receive will not be used for any purpose other than shown in paragraph 2 above; also, it will not be released to anyone without the prior written approval of the Chairman, JSEM Configuration Control Board.

4. I/we guarantee that the provided software package, and/or any modified version thereof, will not be published for profit or in any manner offered for sale to the government; it will not be sold or given to any other activity or firm, without the prior written approval of the Chairman, JSEM Configuration Control Board. If this software is modified or enhanced using government funds, the Government owns the results, whether the software is the basis of or incidental to a contract. The Government shall not pay a second time for this software or the enhanced/modified version thereof. The package may be used in contract with the Government but no development charge may be made as part of its use.

5. The United States Government does not warrant the software package for fitness for use, or any other purpose, or under any other circumstances. The United States Government is neither liable nor responsible for maintenance, updating or correction of any errors in the software package provided. Errors detected in the software package will be reported to the Chairman, JSEM Configuration Control Board, for potential correction.

6. I/we understand that the software package received is intended for domestic use only. It will not be made available to foreign governments nor used in any contract with a foreign government without express written approval of the Chairman, JSEM Configuration Control Board.

7. The JSEM computer program is provided to DoD agencies and other approved users for the purpose of conducting munitions effectiveness and aircraft survivability analyses. Configuration control of this program is the responsibility JMEM Anti-Air Group of the Joint Technical Coordinating Group for Munitions Effectiveness (JTCG/ME). Unauthorized modification of the JSEM program will invalidate JTCG/ME approval of output results; calculations must be repeatable using the currently approved FORTRAN code to warrant consideration for JTCG approval.

Figure 3 JSEM Disclaimer Form

Table 1 JSEM Configuration Control Board

NAME	ORGANIZATION	E-MAIL	DSN	FAX
*KAPPELMAN, Lee	NAWC/418200D	lee_kappelman@imdgw.chinalake.navy.mil	437-3945	437-2062
**WEISENBACH, Mike	WL/XREC	weisenmr@xr_smtp_gw.wpafb.af.mil	785-6262	513 476-7603
CASSIDY, Bill	NAWC/ 418200D	bill_cassidy@imdgw.chinalake.navy.mil	437-3610	437-2062
HANSON, Del	AMSAA	dhanson@amsaa-cleo.arl.mil	298-6369	298-6242
YAROSZEWSKI, Ed	NSWCDD/G24	eyarosz@relay.nswc.navy.mil	249-8851	249-8263
McCOWN, Doug	ASC/XREWA	mccown@eglin.af.mil	872-9589	872-2095
NOFREY, Bruce	NAWC/4KM300E		351-0365	351-0588

Organized by the Missile Evaluation Group
28 June 1989

*Chair
**Co-Chair

2.5 User Support Functions.

There is a pre-processor (JPREP) available and distributed with JSEM. JPREP is a menu-driven, interactive computer program designed to aid in constructing input files for JSEM, modifying JSEM source code, and for sizing internal arrays to optimize JSEM's configuration for a specific problem. JPREP is also an excellent training tool. Procedures implemented by JPREP are designed to configure executable code to the user's specific requirements. JPREP can be bypassed by experienced users who do not need to follow its step-by-step process.

JPREP has the following options:

a. Identifies which files are to be opened when JSEM is executed.
b. Enables the user to build required and optional data input files or to bring together libraries of existing data subsets.
c. Sizes arrays used in JSEM by setting values in the PARAMETER statement.
d. Reduces the list of subroutine names that represent the total executable JSEM package to only those required for a specific run.
e. Prepares the JSEM source code for running on a specific user selected computer and streamlines code if requested.
f. Allows the user to plot target, missle/target encounters, or fragment flyout patterns from the point of warhead detonation.
g. Enables the user who desires to input externally generated fuze points to extract Monte Carlo encounter files.
h. Converts the target contact model from SHAZAM to JSEM format.
i. Provides a review of typical user steps that are normally required to run JSEM.

Within the User Documentation there are listed the phone numbers for Oklahoma State University's (OSU) Eglin Field Office (the model developer) and for the Model Manager to call for help with JSEM. JSEM training is available on demand by calling OSU, although it is not

regularly scheduled. Annual user meetings are held. JSEM has entered into the Survivability/Vulnerability Information Analysis Center (SURVIAC), who are also available to answer questions.

2.6 Assessment and Implications for Use.

The JSEM Configuration Control Process is fairly well structured, and release of new versions is controlled by the CCB. However, the main deficiency of the process is that it is sorely underfunded. While a number of needed improvements in the model have been identified by users and suggested to the CCB, only those improvements which are supported by existing JTCG/ME funding are implemented. This has resulted in several issues remaining unresolved which could affect future users of the model; for instance, the fault tree representation in JSEM has not kept up with the improvements made in the COVART 4.0 version. This inconsistency means that users of COVART 4.0 may not be able to use JSEM for their endgame analysis requirements, depending on whether they have made use of the new fault tree options now available in COVART but not in JSEM.

Another issue is that recent Air Force enhancements to SHAZAM have not been rolled back into JSEM. This is primarily in a change to the input file formats between SHAZAM1 (Eglin version) and SHAZAM2 (WPAFB Version). SHAZAM2 uses FASTGEN 4 formats for target geometry inputs, whereas the previous versions do not. The implications of this change depend on which target geometry is being used to generate vulnerability inputs to JSEM; conversion may be necessary before using JSEM. Alternatively, the new code can be integrated into JSEM fairly easily.

Trouble and change requests go to the Model Manager, who forwards them to OSU. In general, there has been a poor record kept of change notices and documentation updates, and documentation updates have been few and far between.

New versions of code come out with a "what's new in this version" report. Disks are properly marked with version numbers and dates when they are distributed by Tinker AFB.

3.0 Summary of Assumptions, Limitations, and Errors

This section of ASP-I helps the user determine, at an early stage, whether or not a simulation's assumptions, limitations and errors place it outside the realm of applicability to the problem at hand. Coupled with the usage history, the summary of assumptions, limitations and errors can be a powerful simulation selection tool.

3.1 Assumptions

The following are general assumptions inherent in JSEM that may have implications for the potential user:

a. JSEM assumes that the missile has successfully intercepted its air vehicle target (not necessarily hit the target, but come within the vicinity of the fuze detection range.)

b. All trajectories are assumed to be straight lines. This assumption stems from the fact that JSEM is modeling, at most, the part of the missile's trajectory represented by one missile time constant (i.e., the time between the last guidance update and the point of closest approach). In actuality, this straight line assumption only need to be considered to

apply between the first detection of the target by the fuze sensor and the point at which the last fragment from the warhead strikes the target. That time interval is generally on the order of 10 milliseconds. However, for fuzes which make use of guidance provided information, that information must generally be taken several missile time-constants prior to intercept. So while the terminal trajectory can be assumed to be linear for fuzing purposes, the information used by the fuze burst control logic must be based on prior data from the guidance system, which cannot be assumed to be in the linear part of the trajectory.

c. Within JSEM proper, it is assumed that the fuze can be modeled using a right-circular cone approximation of the fuze sensory pattern and the target's detectable surface as a bundle of line segments (Stick-Cone approximation). If that assumption is not applicable to a particular user's problem , the option is available to import an externally generated fuze point from some other source.

d. JSEM assumes 100% reliability of the fuze and warhead.

3.2 Limitations.

The following are some limitations of JSEM as identified by users:

a. The capability to import fuze function data into JSEM (external fuzing) adds flexibility, but only if encounters are generated offline. If the encounters are internally generated, the external fuze mode cannot be used without some manipulation of the files which is not described in the user manual.

b. The "cookie cutter" blast model that is available is rather simplistic.

c. Vulnerable components can only be modeled by spheres or cylinders (for calculation of the number of fragment hits using the expected value approach). When modeling warhead fragment paths using the "ray tracing" approach, all components must be modeled as polyhedrons .

d. For stick-cone fuzing, the target "signature" is not aspect dependent, since it is input as line segments, and the fuze sensor model is strictly geometric.

e. The current version of JSEM (2.2) does not have the capability to have the same component in more than one branch of the fault tree, as is the current capability in COVART 4.

f. The warhead effects model does not consider "synergistic effects" between successive fragment hits.

g. No inherent capability exists to consider variation in the fuze time delay . That is, no error or round-to-round variation in the time delay calculation is allowed.

h. Default values for fragment slowdown are based on a few specific fragment shapes: spherical, cubical, rectangular (only with a length to width ratio of 2), and continuous rods. The user can input the fragment drag function directly, but the documentation doesn't explain how to do that.

i. Structural kill modeling is limited by the availability of appropriate vulnerability models. (generally that is handled using energy density thresholds for structural failure, based on empirically derived data.)

j. A thesis study for the Naval Postgraduate School[5] concluded that a limitation of JSEM is the lack of a central source for target or warhead models.

k. JSEM computes various possible events in a strict order (contact, blast, fragment effects); under certain conditions not all of these events are calculated (for instance, where blast effects kill the target, fragment effects are not considered). This may have some effect on the use of the model, if it is important to keep track of the total effects of all kill mechanisms on the target.

3.3 Errors and Anomalies.

Appendix C contains the most recent errata sheet for JSEM 2.2; this list was compiled as a result of errors found by users and reported to OSU's Eglin Field Office. Most of the errors relate to improper terminations of JSEM, but one or two relate to more serious conditions where the Pk values calculated by the program can be in error under certain conditions.

3.4 Implications for Use.

One of the assumptions associated with the stick/cone fuzing approach internal to JSEM is that the fuze is a narrow beamed "edge detector". When the cone intersects the "stick" target model, that point is assumed to be the first time the fuze "sees" the target. A fuze beam, however, has width. (This type of modeling error can be somewhat overcome by clever use of multiple cones, etc.) However, the actual fuze is sensing the target (accumulating and losing energy as the fuze passes the target.) Time delays applied to the stick/cone detection point can accurately represent actual built-in time delays. However, there is no way for a geometric model to predict the amount of time required for a fuze to "integrate" enough energy to reach its activation threshold. Having said that, it should be stated that geometric models which are carefully based on measurements can be quite accurate.

For high sensitivity, narrow beam RF fuzes, the conical "edge detector" representation has been shown to be a reasonable approximation of the fuze interaction with a target, and for optical systems it is in many cases a very good representation of the optical pattern of the fuze sensor. This is not generally true, however, for threat missile fuzes. Also, for most threat fuzes, there is very little known about the time delays, etc. that make up the burst control logic. In order to develop realistic Pk's, for those systems which have not been completely exploited, a time delay is usually developed for each threat fuze which yields the maximum Pk for an average tactical sized aircraft. This makes the assumption that the fuze is designed for tactical aircraft (e.g., an F/A-18 may be used to define the fuze time delays). This "optimized" time delay is then used against any aircraft. This approach derives Pk's for each aircraft that are reasonable representations of the level of technology available to the threat systems, while not necessarily representing the actual system (which is unknown.)

Users feel that the "stick cone " fuzing model is not a good approximation of semi-active fuzes and is probably not a good approximation of fuzing against LO targets. This may be a significant limitation for a number of users.

[5] A Comparison of Three Computer Weapon-Target Endgame Simulations For Aircraft, Leopold Dolano Gil, Jr, June 1992, Naval Postgraduate School

JSEM's simplistic blast simulation limits its applicability for use with large warheads , or for use against surface targets (for which at least one user has identified a requirement). An outline of an approach for the development of a dynamic blast simulation would call for the inclusion of:

1. Moving charge effects - kinetic energy contributions, blast wave shape, times of arrival…

2. Moving target effects - pressure, vacuum, times of arrival…

3. Blast and frag arrivals - peak damage locations, blast first, frags first…

What is required is a definition of the technical structure for such a model - a meeting of tri-service blast experts to define all the place holders needed to model dynamic blast, discuss related test results and observations, structure model elements for "doable" segments, have place holders for the others, and provide the structure of a model that could be implemented very quickly to replace the current very simple blast appraoach.

A version of JSEM 2.1 was modified by a user to create a version called MOD X which had changes to the blast module and included an AOTD fuze model as a subroutine. This included a version of the SHAZAM method of blast modeling which involves placing a sphere around the burst point - as opposed to placing cylinderical envelopes about the target as JSEM currently does. The SHAZAM blast model may scale better for small warheads. This modification utilizes the polygon surfaces which normally make up the direct hit model to also test for blast kills. This methodology is being modified by another user by:

1. Including scaling for altitude (which is already used for cylinders).

2. Reading the blast radius from the direct hit input file for each polygon so that JSEM can have a blast envelope which varies by component.

3. Including cones, cylinders and ellipsoids in the blast model so that any JSEM compatible direct hit model can be used and not just ones constructed entirely out of polygons.

JSEM input models have become more complex to satisfy user requirements, with the resulting need to speed up some of the code. A case in point is the direct hit routines which have grown from a model of 10's of polygons to models of about 1000 polygons. The current ray tracing process checks for intersections of each polygon surface with each of the 40 rays which make up the missile geometry. This could be speeded up considerably by enclosing the target with bounding box structures. Bounding boxes or spheres are the first order speed up routine used by ray trace algorithms. A second order speedup would utilize some other method such as voxels, bounding box hierarchys, or tree structures such as octrees or BSP (binary space partition) trees.

Fuze models included as subroutines in the code which run off the direct hit skin model geometry and preclude the need for another geometry model for fuzing would reduce the requirement for input data development and the possibilities for error.

Table 2 summarizes recommended fuze models for use with different target types. Note that only the stick-cone model is available as an embedded option in JSEM. For some applications, the stick-cone approximation may be suitable for providing an analysis of trends, where absolute accuracy in the results is not required.

Table 2. Recommended Fuze Models by Target Type

Fuze Type	Target Type		
	Cruise Missile	**Aircraft**	**LO**
Active RF	GTD	Stick-cone (Narrow Beam) PTD (Wide Beam)	PTD
Semi-Active RF	GTD	PTD	PTD
Optical	AOTD	AOTD/Stick-cone	AOTD

4.0 V&V Status and Usage History

This portion of ASP-I summarizes applications JSEM has been used to support, and the extent to which those applications have been supported by V&V documentation. JSEM is the accepted tri-service missile endgame simulation of both the JTCG/AS and the JTCG/ME. As such, it has seen, and is continuing to see, considerable useage within DoD and Industry. In general, there have been no formal V&V activities for JSEM, and what has been done is largely undocumented. There have, however, been a number of related efforts, including sensitivity studies and comparisons with other M&S.

4.1 V&V Status.

This section summarizes any documentation that may be available to support a picture of the V&V status of the model, including studies or programs for which verification and/or validation efforts were conducted, their sponsors, and points of contact, the type of V&V performed, date of publication, etc.

There has been little or no formal verification of JSEM, except for a few unrelated efforts to verify portions of the software. The results of verification of the penetration equations and structural damage mechanism code were reported in a JTCG/ME "Orange Paper"[6].

For the most part, much of what validation has been done is undocumented, and consists of anecdotal comparisons between endgame predictions and actual experience from live missile firings. In general, these anecdotes have reported fairly close comparisons between JSEM predictions and actual test results, but that may be a result of those who are reporting them only being interested in those cases which gave reasonable comparisons.

There have been a number of documented validation efforts for the stand-alone fuze routines which are available for use with JSEM, both for RF fuzes and for Active Optical Target Detectors (AOTD). A model of the DSU-15/B-A/B Target Detector used on AIM-9 SIDEWINDER missiles was developed and documented[7], and validated using real fuze test data[8]. The results of that validation effort led to modifications to the code, after which the fuze point predictions fell within a foot of those measured in the limited test data.

[6] OP-AA/MEG-93-2 Joint Services Endgame Model (JSEM): Integrate and Verify Penetration Equations and Structural Damage Methodology, 30 Apr 93
[7] DSU-15/B-A/B Target Detector Model, Chester Dupuy, NWC TM 4589, 1 July 1981
[8] DSU-15A/B Model Verification Test Data, ASI, Inc., NWC TM 4949, December 1982

For RF fuzes, an extensive validation effort was carried out for the Geometrical Theory of Diffraction code when used against cruise missile type targets[9]. That effort investigated the ability of the GTD code to predict fuze points as compared with actual fuze performance data. Target-fuze interaction data were collected at the Encounter Simulation Laboratory (ESL), located in Corona, California. Two targets and a variety of encounter conditions were used: a Harpoon "training shape", which is identical externally to an operational Harpoon missile; and a "generic" Harpoon which was constructed to conform as closely as possible to the Harpoon shape as modelled by the GTD technique. Runs were made with the Harpoon radome and antenna in various configurations to investigate the effects of the cruise missile guidance system on threat fuze performance.

That effort demonstrated that the GTD fuze model as implemented produced accurate predictions of fuze points for the RF fuzes tested against the Harpoon target. At least 70% of the time, the fuze point calculated by the GTD model was within 2 feet (measured along the missile-to-target relative trajectory) of the actual test fuze point. At a 95% confidence interval, the fuze point was within four feet of that produced by the real system. This is somewhat modulated by the accuracy of the real data: if no data error were present, the study also showed that the difference between real and simulated fuze points is significantly less than that measured. The conclusion of the validation effort was that the GTD model is an excellent way of generating fuzing points and target near-field signatures as measured by the RF fuze, either at a voltage or a power level.

There have been some comparisons of JSEM and other endgame codes. A comparison of JSEM with the endgame codes in ESAMS and MICE II was completed by L. D. Gil, Jr. as a thesis for the Naval Postgraduate School[10]. The conclusions of that thesis related to the ability of each model to assess the kill modes and account for six different vulnerability reduction concepts, (redundancy, component location, passive and active damage suppression, component shielding and elimination). Since the other two models are not really considered endgame simulations, this thesis was not really a validation activity, but it did address the ability of JSEM's functionality to account for those issues. The thesis concluded that JSEM is able to account for redundancy, component location and component elimination. The other concepts (passive and active damage suppression and component shielding) have to be accounted for in the input vulnerability data (these concepts are modeled in COVART).

Another (undocumented) comparison effort was a geometry/parameter check for the fuze simulation contained in JSEM. This effort compared the fuze and blast damage results from JSEM with "hand verifiable" test cases, and with a stand-alone fuze simulation which uses the same basic stick-cone solution. This stand-alone fuze model is an internal NAWCWPNS model with an extensive useage history[11]. The conclusions of this effort were that: (1) the representation of the relative geometry within JSEM is correct (verified by comparison with externally developed results); (2) the fuze model and JSEM use the same relative geometry and target position information; and (3) JSEM gives the correct blast geometry results.

A more extensive comparison was conducted by ASI Systems International and documented in another JTCG/ME "Orange Paper."[12] This effort compared the ROD model contained in JSEM with the corresponding routine in the predecessor program, MECA. Much of JSEM (and MECA) is modular in nature, with an attempt made to keep subroutines to 200 lines or less in length, and make each subroutine's function somewhat intuitive. One notable exception to that rule was the ROD routine, which determines whether the warhead damage mechanism strikes the target, and

[9] Validation of a Geometric Theory of Diffraction Radar Fuze Model, Grant Labarre, NWC TM 6500, Dec. 1989
[10] A Comparison of Three Computer Weapon-Target Endgame Simulations For Aircraft, Leopold Dolano Gil, Jr, June 1992, Naval Postgraduate School
[11] Fuze Definition Methodology: A FORTRAN Code for Simulating Fuze-Air Target Interactions, Carol L. Malone, NWC TP 5137, May 1971
[12] OP-AA/MEG-95-1 MECA/JSEM ROD Model Comparison, 1 Aug 94

where on the target that hit takes place, given position, orientation and velocities at the time of detonation. That particular subroutine was fairly convoluted and difficult to follow. As a consequence, the community's confidence in that particular routine has never been very high. The ASI study was commissioned to delve into the ROD routine, and to compare its implementation in JSEM and MECA.

The results of that study were that a number of coding errors were discovered, primarily in MECA, but a few minor issues were unearthed in JSEM as well. Changes were recommended in JSEM (in the related subroutine CYLIND) which correct potential errors in the expected number of fragment hits on a component, and thus the probability of kill. These corrections have since been made to JSEM version 2.2 This effort was particularly important because of the long-standing issues surrounding the lack of confidence in the ROD routine.

A sensitivity study was conducted using JSEM to examine the accuracy requirements for a number of the variables in JSEM, including fuzing. The sensitivity analysis was performed under the following assumptions:

a. The target was a cruise missile.

b. "True" fuze points were determined by a GTD simulation.

c. A postulated (narrow beamed) advanced threat fuze system was modeled in the GTD simulation.

d. The threat warhead was relatively small.

The relevance of this type of analysis to potential users of JSEM is, of course, directly related to the accuracy of the models used. The GTD model has been shown to be accurate for specific missile size targets under specific test conditions. Within the target size limitations imposed by the GTD code, models using GTD as the nearfield RCS predictor have been capable of predicting fuze function position consistently within one or two feet of measured fuze points. PTD based nearfield RCS prediction codes have been shown to have similar results.

Figure 4 shows the results of the sensitivity analysis. The curve in the figure shows the resulting Pk error (1 sigma) associated with errors in fuzing point prediction (the horizontal axis is error in fuze point prediction, in feet, along the relative missile-to-target velocity vector - Vmt). The sensitivity analysis results shown by the curve in Figure 4 indicate that fuzing predictions within one foot equate to a sigma Pk value of approximately 0.01.

To compare fuzing point predictions from a simple geometric model with the interval requirements shown in Figure 4, a stick-cone model was run for the same set of encounters. Figure 4 also shows the stick-cone derived frequency of fuze detection point errors on Vmt in the various intervals. Examination of this figure shows that less than one half of the fuzing points predicted by the stick-cone model would yield a 1 Sigma Pk value of 0.2 or less (fuze point prediction error of 6 feet or less). That means that over half of the Pk values predicted would have expected errors of $\pm$ 0.2 or greater.

These results are associated with the assumptions and conditions of this particular study. However, this type of result is probably typical of the ability of stick-cone [and other geometric] models to predict actual fuzing points. Often, in defense of using this type of model, the assumption is made that for large targets (for which probability of fuzing is likely to be 1.0) that large populations of intercepts will yield statistically valid overall average Pk values. The validity of that assumption depends on how well the population of intercepts represents the real world population of interest and how well the simplifications of the geometric approach (such as fuze cut-off range) are washed out by averaging over a large number of encounter conditions.

Figure 5 adds the fuze accuracy requirements for both bomber and cruise missile target classes for intercept missiles which have large warheads; if the missile under study (the "air intercept missile") has a relatively large warhead, the fuze accuracy requirements are, understandably, more relaxed than those for missiles with small warheads . However, for a Pk accuracy of 0.01, fuze model accuracies are still on the order of 2 - 4 feet or less.

Figures 6 and 7 illustrate why this is the case. Both of these figures show Pk for various burst positions along the relative trajectory between the missile and the target (which takes the fuze accuracy question out of the picture for the moment) as a function of miss distance. This is for one specific trajectory, rather than averaged over the expected distribution of encounter conditions. Figure 6 is for the large warhead case, and the intervals over which Pk is large (near 1.0) are fairly long (tens of feet), and continue to be long out as far as a 150 foot miss. In figure 7, on the other hand, for the small warhead case, the "lethal intervals" are shorter, and drop to much lower Pk values at 50 - 60 foot miss distances.

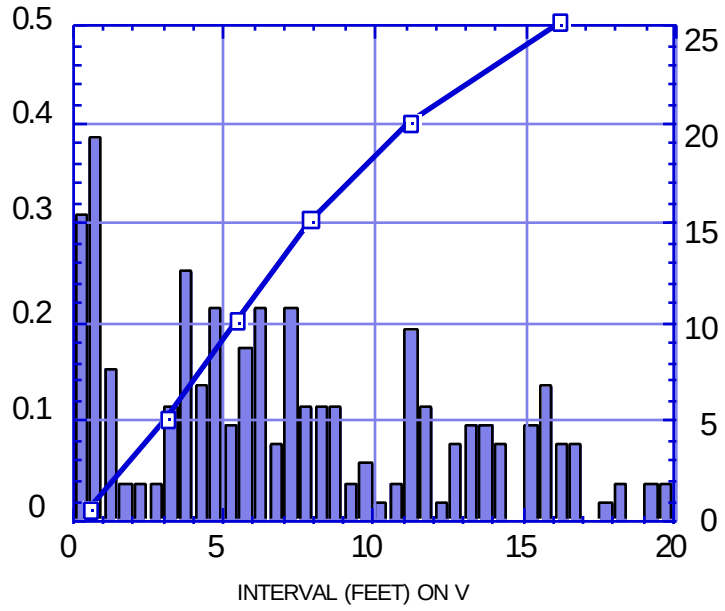

Figure 4 Fuze Accuracy vs Requirements
(Pk Sigma and Frequency vs Accuracy on Vmt)

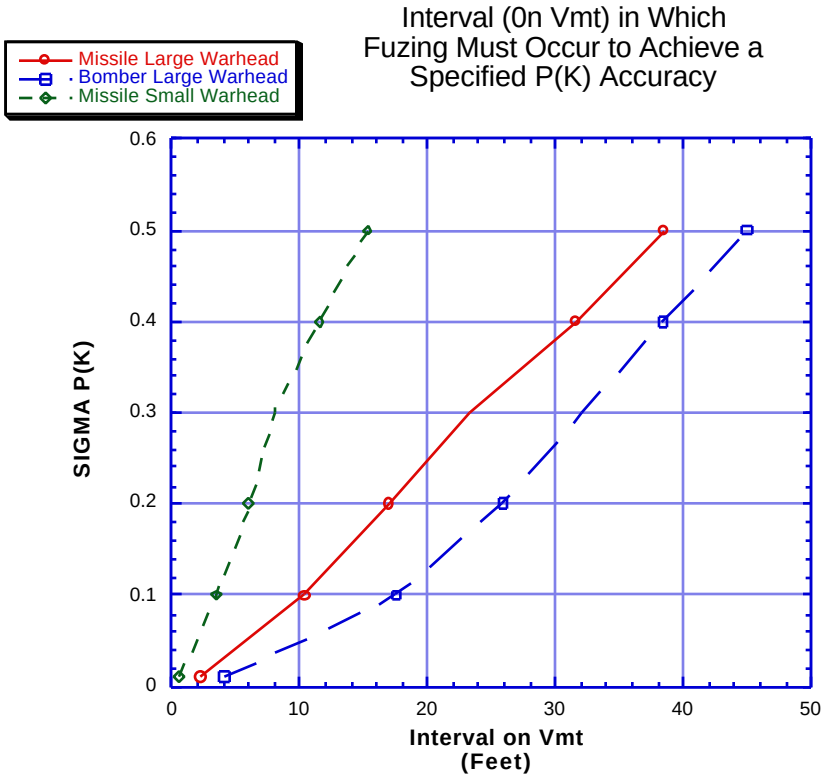

Figure 5 Fuze Accuracy Requirements for Missile and Bomber Targets (Large Intercept Missile Warheads) and Missile Target (Small Intercept Missile Warhead)

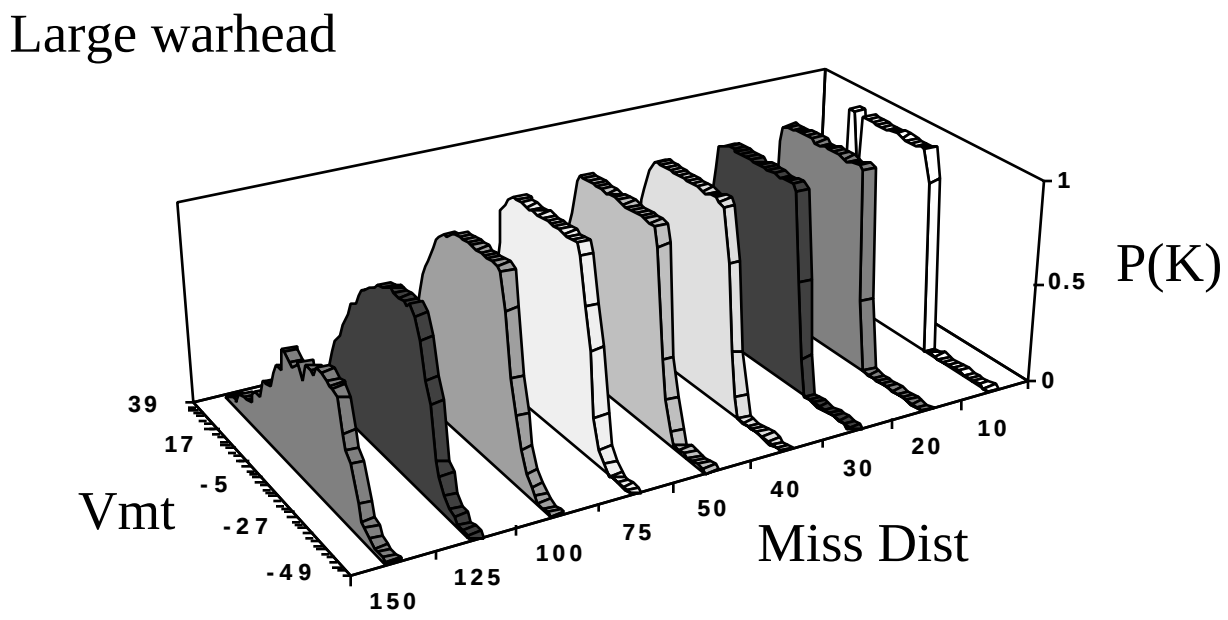

Figure 6 Large Warhead "Lethal Intervals" vs Miss Distance

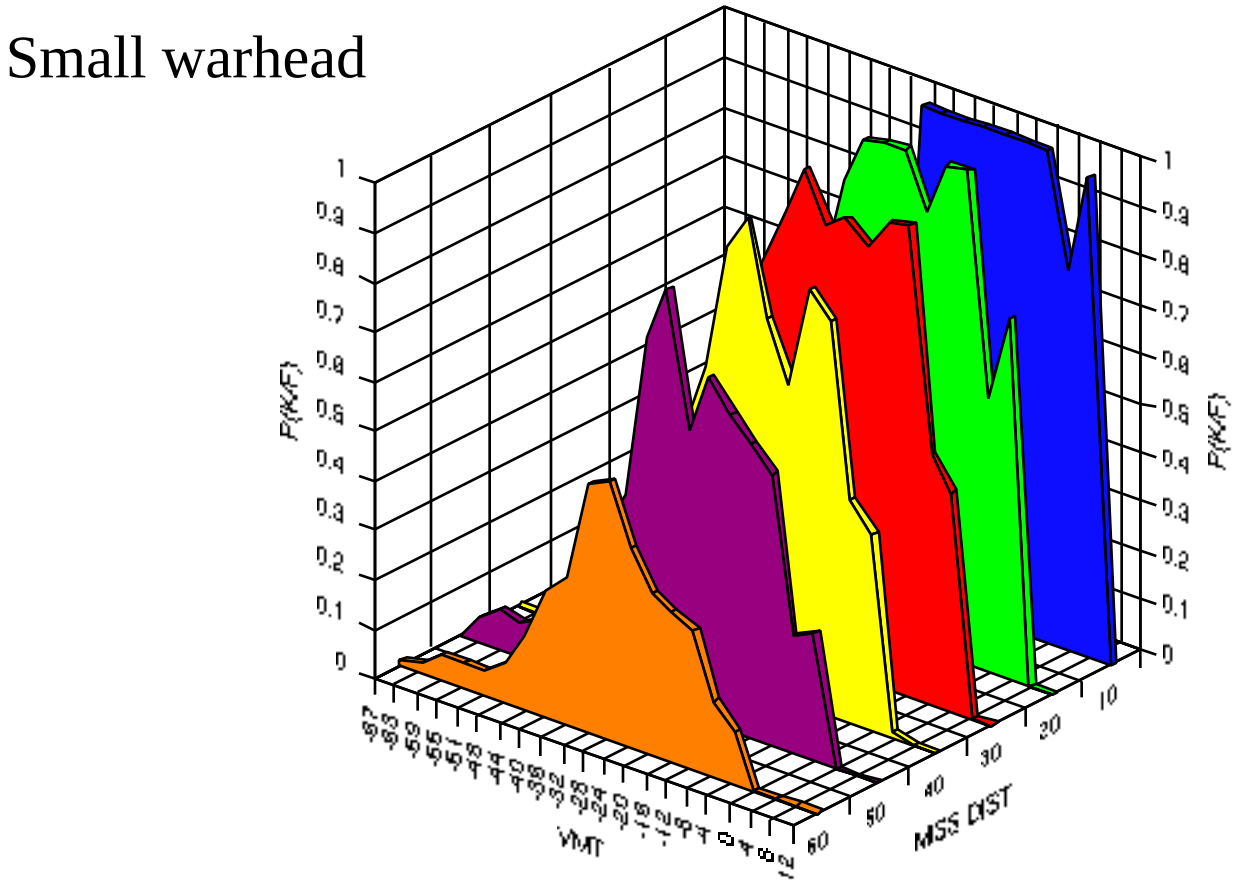

Figure 7 Small Warhead "Lethal Intervals" vs Miss Distance

Table 3 presents a summary of target modeling requirements for fuze simulations versus existing capabilities and the extent of verification and validation (V and V) of the target modeling capabilities. Table 4 shows fuze simulation requirements (based on threats) versus existing capabilities in the two major exploitation/ simulation areas: signal processing logic and sensory response patterns. The extent of verification and validation (V&V) of the threat simulation elements is also included. The Missile Engagement Simulation Arena (MESA) at China Lake is a fuze test facility which expands on the capabilities that were available at the older ESL in Corona.

Table 3. Target Modeling Requirements for Fuze Simulation vs. Existing Capabilities

REQUIREMENTS: TARGET TYPE	EXISTING CAPABILITIES	EVALUATION/CONFIDENCE IN EXISTING CAPABILITY AND EXTENT OF V&V
Small/missile size targets that can be described using primitive geometric shapes	**Simple geometric models:** • Stick-cone (resident in JSEM) • Glitter point	These types of models are not adequate primarily because P(F/I) is assumed to be 1.0 which may be an incorrect assumption for missile size targets. These types of models have been validated only for specific targets vs. specific fuzes. The validation is of limited use for more generic models.
	Nearfield RCS Prediction Code: • Geometric Theory of Diffraction **(GTD)**	The GTD code is an excellent nearfield prediction code for small/missile size targets. The Ohio State originated code has been validated with measurements for primitive geometric shapes and missile size targets Within the target size limitations imposed by the GTD code, models using GTD as the nearfield predictor have been capable of predicting fuze function position within one foot of meassured fuze points. Analysis documented in Figure 4 indicates that fuzing predictions within one foot equate to a Pk error sigma value of approximately 0.01
		GTD is a very efficient and fast running code which makes it attractive for fuze simulations. Fuze simulations tend to be very computer intensive.
Large/complex shape targets that cannot be reasonably described using primitive geometric shapes	**Nearfield RCS Prediction Code:** • Physical Theory of Diffraction **(PTD)**	PTD has most of the same attractive qualities that GTD has with the additional benefit that all size targets can be modeled. Large targets can often be adapted from existing CAD models. PTD has validation accuracies similar to GTD. At this time PTD has a major drawback: it is very computer intensive and long running. For that reason GTD is preferred (at this time) for modeling small targets in fuze models that are to be used in high production modes.
	Simple Geometric Models • Stick-cone (in JSEM)	These types of models are only adequate for narrow-beam, high sensistivity fuzes. In those cases, fuze accuracies may be sufficient for many types of lethality and/or survivability analyses.

Table 4. RF Fuze Modeling: Requirements (to Correct Deficiencies) vs Proposed Enhancements

REQUIREMENTS TO CORRECT DEFICIENCIES	PROPOSED ENHANCEMENTS	STATUS OF ENHANCEMENTS
RF Active and Semi-active Fuze Simulation • Nearfield RCS • Small target	**GTD Update:** • Update existing GTD simulation * Include semi-active geometry * Current threat expliotation (specific threat fuze logic) * Current GTD code development * Include various target materials * Streamline simulation * Validate with MESA testing	GTD is the only nearfield code (at this time) that has run times that are reasonable for any sort of production mode fuzing prediction. This type of simualtion is absolutely necessary for missile size targets. GTD upgrade tasks are being worked on by two missile projects. Due to funding cut-backs, the tasks are proceeding slowly. Additional funding is required.
RF: Simple Fast running Fuze simulations • Reasonably accurate • All size targets • All current threats	**FAST FUZE MODEL Development** • Develop fast running fuze simulations * Based on high fidelity codes and testing * For use in high production modes * Include P(F/I) * All target sizes * All current threats	This type of model is not currently available in the community. The need for such a model (or set of models) is painfully obvious to those of us working with the limited set of current fast running models. To be an improvement, FAST FUZE would have to be based on high fidelity nearfield codes and validated with test data.
RF Active and Semi-active Simulation: • Nearfield RCS • All size/shape targets	**PTD Development:** • Update existing PTD code * Include semi-active geometry * Current threat exploitation (specific threat fuze logic) * Current PTD code development * Include various target materials * Streamline simulation > Use GDT approach to start calculations * Validate with MESA testing	The PTD code is currently being developed by the Electromagnetic Magnetic Consortium (EMC) and a missile project. Ultimately PTD (or a similar code - maybe a combintaion of GTD and PTD) is the solution for calculating nearfield RCS for all size/shape targets. The PTD approach at this time is unwieldy for the current generation of computers (run times are way too long). We need, however, to prusue the development of codes that work for all target sizes and shapes. Simplier fast running codes can be developed using the GTD and PTD codes.

4.2 Usage History.

This section summarizes the history of prior uses of JSEM based on inputs from user surveys, model managers, model developers, government agencies, and other pertinent sources of such information. It includes a description of the types of application for which JSEM has been applied.

JSEM has seen considerable use over the last several years for a number of applications. It is the standard missile endgame tool for both lethality studies (including JTCG/ME official publications) and survivability studies (it is now the accepted JTCG/AS endgame model, and it has been entered into SURVIAC).

In support of survivability analyses, JSEM serves as the principal tool for development of Pk's of threat missiles against domestic air vehicles. An example of the type of information generated in support of survivability studies is missile "Lethality Envelopes," or "Pk given launch envelopes." Pk given launch envelopes are generated for surface to air (SAM) systems against many aircraft. These envelopes consist of locations in a grid about the SAM site where successful launches occur, for straight and level fly-by of the SAM site at a specified speed and altitude. Within the launch envelope, at each point on a predetermined grid, the average Pk is generated using JSEM and plotted over a number of (typically 10) flyout iterations of the model.

For Air to Air missiles (AAM) a similar procedure is accomplished, only in this case the results are generally known as Launch Acceptability Regions (LAR), with Pk's plotted within the LAR. These results are generated for a number of air target maneuvers, and generally they are plotted with reference to the target at the center of the plot, where launch positions are shown for the shooter within the plot (rather than shooter centered, as is usually the case for SAM envelopes). The Joint Munitions Effectiveness Manuals (JMEMs) for US AAM are generated in this way using JSEM for Pk development.

Some of the previous uses for JSEM as reported by members of the User Group are as follows:

- Support of design optimization/verification studies leading to improved weapons designs and enhanced combat capabilities for US and Allied forces
- Support of survivability and vulnerability analyses
- Air target modeling and simulation of signatures; tech base research
- A tri-service approved method to evaluate CORPS SAM missile lethality for targets or variants not currently contained in the ELEGS/PEELS databases.
- Provides tri-service approved method for missile endgame effeciveness assessments for advanced concept developments
- Enhances simulation capability under the PROTEC program
- Used on the AIM-9X program and other weapons studies
- Used as a design tool by missile programs
- Used by acquisition programs in support of COEAs
- Used to support system safety risk assessments and evaluation of warhead concepts
- Will be used in conjunction with an integrated vulnerability workstation (database for vulnerability models)
- JSEM is the JTCG Standard for conducting endgame assessments of US and foreign aircraft.
- JSEM is used to produce effectiveness data as tasked by the JMEM/AA working group
- Conducting lethality studies for US missile programs
- Assess lethality of SAM systems against JSOW Baseline, BLU-108 and JSOW Unitary weapon
- Used to provide endgame data for use with ESAMS and BRAWLER models, in support of concept definition and system trade studies
- Used in support of surface to air gun effectiveness analysis
- Used to help in analyzing fleet missile firing performance, and to aid in selection of scenarios for fleet missile firings
- Being evaluated as a potential endgame program to compare actual combat damage with JSEM runs to aid the vulnerability community
- Used to evaluate existing and prototype fuze concepts
- Evaluation and potential development of superior air defense system
- Used to provide insight into the vulnerability of the B-2 aircraft to conventional SAM encounters
- Used to provide higher fidelity Pks than alternative approaches
- Used to evaluate JASSM warhead design

- Used to validate hardware-in-the-loop simulations
- Used to model the effectiveness of foreign warhead designs
- Supports MICOM RDEC Structures
- Supports the conduct of fuze and warhead design studies

4.3 Implications for Use.

JSEM is used by a wide variety of users across government and industry. It is the standard tri-service missile endgame analysis tool for both survivability and lethality, accepted by both the JTCG/ME and the JTCG/AS. Uses of JSEM have included US anti-air missile fuze and warhead design studies, evaluation of threat system effectiveness (missiles and guns), support to operational as well as development testing, and evaluation of design alternatives in COEAs and system design trade studies. The implication of JSEM's useage history is that no other endgame model would have the equivalent level of credibility or acceptance for these types of application.

5.0 Documentation Assessment

This section reviews the current status of JSEM's documentation with respect to standards developed for the verification of mature M&S. The standards were developed by reviewing MIL-STD, DoD-STD, JTCG/AS and service specific policies, procedures and guidelines relating to M&S development, and tailoring these standards to the problem of "V&V in reverse" for mature M&S.

Currently available documentation for JSEM are the following:

Joint Services Endgame Model (JSEM) Computer Program, Version 2.2 (Magnetic Diskette), 61JTCG/ME-88-3 (source code and sample cases) 2 August 95

Joint Services Endgame Model (JSEM) Computer Program, Volume I - An Introduction to JSEM (Getting Started), 94 pp., 61JTCG/ME-88-3-1, 30 June 95. This volume is an introductory guide for the new or infrequent user of the JSEM computer program and the preprocessor program, JPREP. It contains an overview of both JSEM and JPREP. JSEM input files are required to run the program, and the output files are discussed. Debugging and sample cases are also included.

Joint Services Endgame Model (JSEM) Computer Program, Volume II - Reference Manual, 396 pp., 61JTCG/ME-88-3-2, 30 June 95. This volume is a reference manual for those familiar with running JSEM. It contains instructions for building JSEM input file records and for sizing arrays using the PARAMETER statement values that are keyed to specific input of the program variables. Information is provided to assist in selecting the appropriate program modules to accomodate desired damage assessment methodology.

Joint Services Endgame Model (JSEM) Computer Program, Volume III - Analyst Manual, 61JTCG/ME-88-3-3, UNPUBLISHED. This volume contains mathematical explanations of the JSEM code.

Other related documents which are available through the JTCG/ME include a number of "Orange Papers" documenting various studies. These and the above JSEM documents may be obtained through the OSU Field Office at Eglin AFB, (904) 882-5554.

OP-AA/ATV-90-1 Missile Debris Sensitivity Analysis, 9 Jan 90

OP-AA/ATV-92-1 Sparrow Missile Debris Characterization, Feb 91

OP-AA/ATV-93-6 A Proposed Solution for the COVART/JSEM Problem, 22 Feb 93

OP-AA/MEG-93-2 Joint Services Endgame Model (JSEM): Integrate and Verify Penetration Equations and Structural Damage Methodology, 30 Apr 93

OP-AA/ATV-94-2 Missile Debris, 11 Feb 94

OP-AA/MEG-94-1 Target Probability of Kill: A Discussion of the Joint Services Endgame Model (JSEM) and Aircraft Vulnerability Analysis, 10 Nov 93

OP-AA/MEG-94-2 Subroutine ROD: A General Damage Mechanism Impact Model, 15 Sept 94

OP-AA/MEG-95-1 MECA/JSEM ROD Model Comparison, 1 Aug 94

OP-AA/MEG-95-2 Joint Services Endgame Model (JSEM) Update to Accept the Output File from COVART 4.0, 5 Oct 94

5.1 Completeness.

The User Manual is basically complete and useful. The Analyst Manual, however, is sorely deficient (only an unpublished draft exists, which is not available to most users unless they request it of the Model Manager). The "Getting Started" manual is designed for the new user and is generally useful. There is no software design documentation (Conceptual Model Specification), even in the draft Analyst Manual.

5.2 Compliance.

The User Manual complies with the accepted standards. The Analyst Manual, even in draft form, does not satisfy the requirements , and there is no software design documentation.

5.3 Recommended Modifications.

Needs an Analyst Manual and Conceptual Model Specification. The User Manual needs to be updated with each new version release.

5.4 Implications for V&V.

The lack of an Analyst Manual and a Conceptual Model Specification (or Software Design Document) would make any verification effort extremely difficult, if not impossible. Verification acitivites would have to begin with the generation of both those documents. The configuration management process as it exists would support the long term value of any V&V efforts, but such efforts would require considerable preparatory work.

5.5 Implications for Model Use.

The User Manual provides enough information to provide a user with the information needed to make the model run, and JPREP does an excellent job of assisting the user with setting up runs. However, there is no information available to tell a user what the code is really doing. As a consequence, it would be extremely inadvisable for a new user to try to perform analyses using the code without detailed and extensive help from someone experienced with the code.

6.0 Software Quality Assessment

This section gives the prospective model user an indication of the conformance of model code to accepted software development and documentation practice. The structure of the source code of a given model is analyzed from a software engineering perspective in three major areas: use of programming standards; computational efficiency; and memory utilization.

6.1 Programming Conventions

JSEM is written in ANSI standard FORTRAN 77. Since it serves as a Joint Service code, it was expressly developed with the requirement that there be no machine specific elements. Thus it is extremely portable, especially with the assistance of the JPREP pre-processor.

The goal during development was to keep subroutines at less than 200 lines of code. For the most part, that goal was achieved. Other efforts to enhance portability included keeping variable names at 6 characters or less.

6.1.1 Use of Embedded Comments

An attempt has been made to embed comments throughout. Some subroutines are dated with their most recent modification date, but this convention started in the MECA code was not consistently continued in JSEM. The use of embedded comments is fairly consistent and informative. There could be more comments, since some of the older code does not have any explanation.

6.1.2 Use of Module Preambles

A shortfall of the code is that no subroutines have classification markings (probably due to the fact that the standard version of the code is unclassified). Many subroutines begin with comments containing definitions of the variables used in the subroutine, but this again has not been continued in JSEM. Comments in many subroutines also list which subroutines it is called by, and which subroutines it calls. This also is not consistently done throughout JSEM.

6.1.3 Source Code Formatting

The code generally follows indentations. Since this was written by committee, there are a number of widely differing programming styles evident in the code. There is a book of flowcharts as part of the analysts manual (which is unpublished). No work is currently planned or budgeted for updating the analyst manual.

6.1.4 Logical File Processing

The JPREP preprocessor will allow the user to configure the program to the user's computer equipment (open statements, etc.). JPREP is where a user builds the "parameter file", which allows the user to change the array sizes internal to JSEM, depending on the size of the vulnerability model, warhead model, etc.

A subroutine called OPNFIL allows the user to assign any name to the various I/O files used by the program. This is done as part of the input stream, or "run script." However, the code requires the user to know the file number assigned to each specific I/O file. For example, the user inputs the file name for "unit 21", having to know that unit 21 is the encounter input file. This allows for potential mis-matches between unit numbers and I/O files. A much better approach would be for the "run script" to call for the name of the "encounter input" file, and assign a number to it internally.

6.1.5 Variable Declarations

Volume II of the User Manual contains a comprehensive list of variable names and definitions. These names generally relate somewhat to the function of the variable, but this is not always consistently done throughout the software. JSEM does not generally declare variables explicitly, but rather falls back on the default FORTRAN conventions. Only those variables which run contrary to the default convention are declared. This has not caused any execution problems, including on machines which normally expect declaration of all variables.

6.1.6 Programming Logic

Since JSEM's subroutines are all small, they are concise and solve a very specific problem. There are no compiler or hardware dependent elements. There are very few local variables: all are global through the use of common statements. Subroutines generally have an error exit (for exceeding maximum dimensions, for example - each subroutine checks input values against what's expected and errors out if it doesn't match). Other than the error exit, there is generally only a single normal exit point from each subroutine. If JSEM finishes abnormally, it will tell you where (and in some instances, why). The error messages from JSEM are, for the most part, informative and useful.

6.2 Computational Efficiency.

An assessment of computational efficiency provides a description of the code elements that affect efficient implementation and/or execution of the software. This may be a point of minor interest depending upon the user and the intended target computer system for the model, but machine dependent aspects of the implementation can significantly affect its use. The following factors examine the efficiency aspect of the actual coding: modularity, algorithm development, and variable allocation.

6.2.1 Modularity

The code is very well organized. Within each module, it is easy to follow the flow. This is not so much true for the code overall, mostly due to the lack of documented flowcharts, but for specific error checking and to determine why an execution failed, subroutines are easy to understand. Comments at the beginning of each subroutine support what it does, but internal to each

subroutine (where there may be few comments sprinkled about) it may not be obvious what it's doing.

6.2.2 Algorithm Development

It is difficult to assess the implementation of the algorithms and their efficiency without an analyst manual. Some of the algorithm development is lost in the mists of time, and nobody really knows where it came from. This is a major shortfall of the code.

6.2.3 Variable Allocation

Global variables are used through common statements, which seems to work for JSEM.

6.3 Memory Utilization

Efficient use of memory by the software has become less important than other quality factors due to its declining cost and increasing availability. Memory use, however, can be a critical factor for some applications that may also be restricted to a particular type of processor and its associated operating system.

6.3.1 Global Memory

No EQUIVALENCE statements are used in JSEM, but there is widespread use of COMMON statements to make global variables. JSEM is not a huge code, so memory is generally not a problem, and the JPREP pre-processor allows the user to configure the code to whatever memory requirements he or she has.

6.3.2 Local Memory

N/A

6.4 Implications for Model Use

The assessment of the users is that "JSEM is OK" They give it a grade of "B" for software quality related factors. For the most part, JSEM is modular and does not have long subroutines which are run together with lots of "GO TO's" (with a few minor exceptions, it is not what would be considered "spaghetti code"). However, it was written by committee, so it could be prettied up, but functionally it works and is understandable. Nesting of loops may not be the most efficient. It could be consistently indented, commented, etc. and generally modernized. But there are no serious implications for use based on software quality issues.

APPENDIX A

SUMMARY OF CHANGES TO JSEM VERSION 2.1

Listed are the changes made going from version 2.1 to version 2.2 of JSEM. Modifications were functionally grouped and changes are generally noted in the right-hand column of the JSEM source code.

1. IMPLEMENT COMBINED METHODOLOGY .

Combined methodology allows the entry of two distinct warhead models. The first one represents the actual warhead and is evaluated using ray-trace methodology. The second one represents small missile debris fragmentation and is evaluated using expected value methodology. When using this type of study, each component must have two separate kill mechanisms (one for each "warhead") and the geometric representation of the components must be in polyhedron format.

2. SURFACE TO AIR GUN MODIFICATIONS.

This modification allows JSEM to perform Surface-to-Air Gun studies similar to program PKD. Satisfactory testing for this modification would have unduly delayed the release of version 2.2 so full implementation has been withheld until a later time. However, there is code present in version 2.2 intended for this purpose including the implementation of a PKD type scalar that can be assigned to each component (This is generally a function of the intercept range for point defense weapon).

3. IMPLEMENT ARRAY STORAGE FOR FFCM STUDIES.

These modifications are to implement array storage instead of direct access files for NMT studies. The use of direct access files caused contradictory answers on different machines due to varying machine word lengths.

4. INCORPORATE MECA ENCOUNTER PARAMETER CALCULATIONS.

These modifications assure that all necessary encounter parameters are calculated. Most of the modifications came directly from the latest version of MECA. Note the following changes:

a. The supervisor routine ENCADD was added to properly administer the parameter calculations.
b. Subroutine CHIZTA was replaced by ATTANG; computes the angles chi and zeta.
c. Subroutine ANGATT was replaced by YAWPIT; computes missile yaw and pitch.
d. Subroutine ATTANG was replaced by THPXMU; computes theta and mu.
e. Subroutine DETSPN was added to calculate the vector from the burst point to the aimpoint; now called from MAIN rather than BLSTPK. (The possibility of errors existed in version 2.1 if the target did not contain a blast model.)
f. An error was detected in subroutine CMPPK. The JSEM speed-up code would not key on the spanning node for a spherical component and potentially lethal polar zones were not being evaluated. This problem was corrected.

5. INCORPORATE MECA INTERNAL FUZING MODIFICATIONS.

Changes were made to incorporate improvements in the internal-fuzing algorithms made by the MECA users group.

6. INCORPORATE MECA FAULT TREE MODIFICATIONS.

Modifications involved changes to the way the fault tree is parsed and variables are stored. (The actual evaluation algorithms remained the same.) Also, more informative error messages are now issued.

7. IMPROVE DIRECT HIT EVALUATIONS or (DIRHIT).

Changes were made to allow JSEM to implement contact fuzing. In Addition, JSEM now evaluates the first surface struck by each missile contact point (rather than only the very first surface struck by the missile as in version 2.1).

a. Two new variables are used to implement the contact fuzing capability. Refer to User Manual records CONT-3, CONT-6, and CONT-8 for details of the new fuzing flag associated with each surface. Also, see record FUZE-2 for a description of the time delay associated with a contact fuzing.

b. The final report block in the SUMPRT file was updated to reflect changes in the direct hit evaluation. It now contains the number of contact fuzings initiated. A major difference between the new report block and the one found in version 2.1 is the definition of contact and debris kills (2.1) and contact and debris evaluations (2.2).

c. For version 2.1, each number in the contact kill column represented an encounter where:
   (1) The first surface struck was described as 'CONT' vulnerable.
   (2) The first surface struck had an assigned Pk of 1.0.
   (3) Impact occurred before proximity fuzing. (The intact missile struck the target.)

d. For version 2.1, each number in the debris kill column represented an encounter where:
   (1) The first surface struck was described as 'DEBR' vulnerable.
   (2) The first surface struck had an assigned Pk of 1.0. Note for version 2.1, a kill occurred before it was recorded in either column.

e. For version 2.2, each number in the contact evaluation column represents an encounter where the first impact occurred before proximity fuzing. Each number in the debris evaluation column represents an encounter where proximity fuzing occurred before the first impact. (All direct hit damage is due to large missile debris.) Note that for version 2.2, a single encounter may be recorded in either of the above columns and (if the direct hit Pk is 0.0) may also appear in either the blast kills or fragmentation evaluation columns.

8. EXAMINE AND DEFINE DYNAMIC CONDITIONS THAT CAUSE CATASTROPHIC ERRORS.

These corrections detect and define conditions (generally caused by specific polar zone boundary/component orientations) that would result in a divide by zero error causing a nongraceful program termination.

9. STANDARDIZE INTERNAL COMPUTATIONAL UNITS.

Originally intended for version 2.2; but, due to time considerations is being withheld for a future version of JSEM.

10. IMPLEMENT A CONFIDENCE LEVEL TERMINATION GIVEN EXTERNALLY GENERATED ENCOUNTERS .

This change requires three new input variables. (Refer to Users Manual record PCNT-1 for details.)

11. MISCELLANEOUS MODIFICATIONS.

These include:

a. A number of hits column has been added to the AVPK file output.
   (1) If a ray-trace study, the actual number of fragments striking the component is indicated.
   (2) If expected value, the number of zones evaluated against the component is indicated. NOTE: one zone given by specific ejection angles may be counted several times based on the number of different fragment weights, ejection velocities, etc.
   (3) If combined methodology, the number of hits is a combination of the above.

b. The encounter number has been added to each fragmentation flyout file and the maximum number of fragments that can be graphed has been increased from 999 to 99999.

c. The implementation of a scalar multiplier to limit the total Pk contribution of a FIRE vulnerable redundant group. (See User Manual record RDNT-4 for more information.)

d. Informational messages concerning which nodes a polyhedron were reduced to (for an expected value study) and when a fragment mass or velocity is outside a table value will only be written when IOPRNT is set to a '2' or '4'.

e. Corrected an error in the ray-tracing speed-up code that would occasionally neglect to look in a radial area that might contain vulnerable components.

f. For a specific trajectory evaluation with multiple-detonation points, the last point was being evaluated twice and the target Pk would be accumulated twice. This was corrected.

g. Corrected an error for NMT (FFCM) studies that involved the angular offset between warhead rings.

h. Subroutines FIREIN and WIZBAN were changed to allow a kill mechanism of 'FIRE' to access a quadruple-interpolation table. See User Manual record FRAG-21.

i. Modified the confidence level checks for both Monte Carlo encounters and the warhead roll Monte Carlo.

j. Improved messages written to the AVPK file output.

APPENDIX B

CHANGES/MODIFICATIONS/UPDATES TO JSEM 2.2.2

as of 20 May 1996

Blast Damage: An option has been added to the code which allows the user to input either hemispherically-capped cylinder data or static blast radius at sea level value. The cylinder data remains the same as before. The static blast radius (feet or meters) is entered on warhead record two (WHRD-2). This value is the distance over which the blast over pressure will cause catastrophic damage to the target. NOTE: If the static blast radius is used, the direct-hit model (target skin) MUST be described by polygons and the direct-hit and the damage for each polygon MUST be `DEBR`.

Code in subroutine XVALPK. There are only five warhead options: `AIMABLE`,`FFCM`, `FRAGMENT` ,`SPOKE`, AND `CR`. Therefore, changed `CONTROD` to `CR` and deleted code which dealt with TYPWHD .EQ.

Deleted subroutine LENGTH. This subroutine was only called if TYPWRD .EQ. ` ` in subroutine XVALPK. As this type of warhead is not one of the allowable choices, the subroutine can be deleted.

Delete code in subroutine DETAPC. Subroutine DETAPC is called only if the type of detonation position option desired, TYPDET, is set to `SPECIFY`. Therefore, all code referencing `CONTOUR` was removed from subroutine DETAPC.

Write all error messages to DTLPRT (detailed output file). The summary output file should only contain a summary of the JSEM computer run. Therefore, the code was modified to write all error messages to the DTLPRT. This also complies with the print option flag, IOPRINT.

Add mode encounter 6 (MODENC6). China Lake requested that a new encounter based on launch range be added to the options available in JSEM.

Read an ESAMS output file. The JTCG/AS requested that JSEM/JPREP be modified to read an ESAMS output file. It was intended that after this modification, this version of JSEM (version 2.2.1) would be placed into SURVIAC.

Add GRFIL to the list when opening all files. Present code would open all but the GRFIL file; code was modified so GRFIL file would be included in "all files" in JSEM subroutine OPNFIL.

Convert JSEM and JPREP to run in Windows environment. Both JSEM and JPREP have successfully run in the Microsoft Windows environment.

Option to print encounter number to screen/log. Old code would print "starting encounter no. XXX" to the log file or to the screen; where xxx was divisible by five. This resulted in large numbers of messages to the screen/log file if the computer run involved a large number of encounters. With the new code, the user enters an integer, NUMENC, on record PCNT-5. If a "0" is entered, no message is written to the screen/log; otherwise, the encounter number evenly divisible by NUMENC is written to the screen/log .

Normal termination if ENCFIL and DATFIL do not agree. With the old code, if the maximum number of encounters to be evaluated, MAXENC on record PCNT-5, was greater than the number

of encounters in ENCFIL (encounter file), JSEM would terminate when attempting to run the next encounter past the end of the ENCFIL. With this change, JSEM will complete the run successfully and write a message to the screen/log that the number of encounters requested exceeded the number in the ENCFIL file.

Initialize variable IWRCP1 to 10. If only one vulnerable component and only one fragment were being evaluated, and the user wished to plot the fragment fly out; then the screen would not properly size. (This would only occur during debug or stepping through the code.) Code was modified to accommodate all number of components and fragments.

Option for either TDFFIL or AVPKCP time for plotting missile/target encounter. With the old code, the user could only select the time of the encounter from the TDF file, the scheduled time of detonation. This is the actual time of detonation unless the intact missile intercepts the direct-hit model. If an intact missile intercepts the target and the missile detonates, then the program recalculates the detonation time and writes this new time to the AVPKCP file. If the missile does not intercept the target, the time of detonation written to the AVPKCP file is the same time as that contained in the TDF file. Within the modified code in JPREP, the user can select the scheduled time of detonation in order to examine the relative positions of the missile and the target before the production run begins: or the user can use the actual detonation time(from the AVPKCP file) to plot the encounter after a production run is completed.

Ensure PKTAB file header is always read. Again , if only one vulnerable component, then the PKTAB was not read. With the modified code the PKTAB file header is read regardless of the number of components.

Write PARAMETER only once. Reformatted PARAMETER statement to four variable across vice three variables across; thus was able to fit all variables onto 16 lines (before, 21 lines were required for all variables and the word PARAMETER was inserted twice to avoid the computer default setting of 19 continuation lines (the HP computer bombed with the two PARAMETER words.)

Rewrote total target PKPRT. If KLLCNT = 1 (process all damage mechanisms) then the target PK could be greater than 1.0. The old code wrote the total target PK as if only one damage mechanism had been evaluated; however, the sum of the Direct Hit, Blast, and Fragmentation PK could total more than 1.0. The modified code places asterisks in the place of total target PK and writes a descriptive message directly below the summary table of the SUMPRT.

Delete/Modify code in subroutine KLMOUT. Old code referenced "Probability of Damage" which was changed to "Probability of fuzing"; and code dealing with PKIMPT (never used in JSEM) was deleted.

Standardized header and TGTGS in all encounter subroutines. Code, added to check for target "Gs" in all encounter subroutines and headers, was changed so as to read the same.

Deleted subroutine CNTLRD. This subroutine dealt only with 60 grain control rod fragments (a special study done years ago at China Lake) as is no longer needed (code was archived).

Deleted reference to MFRAG. Code originally written to address multi-fragments: JSEM deals only with single shot (single fragment) PK; thus code was deleted and archived.

READ.ME file added. A READ.ME file was added to the disk in order to assist users in extracting JSEM files from a 3.5" disk to their production machine.

Change code in subroutine AVOUT. With old code, if KLLCNT = 1 (JSEM to evaluate all damage mechanisms) and IBLAST=1 (target killed by blast) then information about vulnerable

components hit was not written to the AVPKCP file. New code permits writing component data to the AVPKCP file.

Rewrite computed GOTO statements. Computed GOTOs are difficult to trace; thus all computed GOTOs were rewritten to IF-THEN- ELSE statements.

Clarify FLTTIM. Added additional notes on record PCNT-2 to clarify that FLTTIM (constant fragment fly out time) is only used if all fragments are to be plotted.

Initialize control record flags. Good programming practice would be to initialize variables; thus all control record PCNT) flags are initialized to a default value.

Set PI=3.14 in subroutine POLCHK. Round off error was causing some polygons to be labeled concave, changing PI to 3.14 avoids this problem.

Change NINT to NINTA. NINT is a reserved word; thus the variable was changed to NINTA.

Update sample case matrix. Matrix was modified to indicate that fragment fly out (FF) is a plotting option with the combine warhead.

Rewrite subroutine SHAMEC. Subroutine rewritten to build FRAG (vulnerable components) section and CON (direct hit model) from SHAZAM tape 11 and to build PKTAB file from SHAZAM tape 12. Subroutine was renamed SHZJSM (SHAZAM TO JSEM).

Add check to PKTAB file. Code modified in subroutine PKFIN to read PKTAB file header record and to check that the correct number of records are present in the PKTAB file. Comments were added to clarify single, double, triple, and quadruple PK tables.

Modified message in JPREP. A more readable message in JPREP subroutine MDP when generating JSEM source code.

Remove variable NVRTCS from PARAMETER statement. Variable NVRTCS (maximum number of vertices) is not used in JSEM; thus removed from the PARAMETER statement.

Renumbered/Insured that T88 code present in all JSEM subroutines. With deletion of several subroutines, the remaining subroutines were renumbered. Also, code was added to insure that all JSEM subroutines have capability to write information to TAPE88 file (subroutine was called by program at least once.)

Update JPREP subroutine T88MKR. Insured that subroutine could build JSEM source code tailored to a particular DATFIL (data file).

Substitute RANGI for RTGDIS. China Lake requested that RANGI be substituted for RTGDIS. RANGI has the same definition and is used in both the RANG section of the DATFIL and in the M6ENC encounter (MODENC=6)

Add error check for CONTACT FUZE INDICATOR. Added code to check that contact fuze indicator for all three Direct-Hit shapes is set to either a "0" or a "1".

Add note to VATB and XVAL components. Added note that vulnerable components with VATB measure of vulnerability are listed before vulnerable components with XVAL measure of vulnerability in the DATFIL (data file).

Added note to FRAG-3 and RDNT -3 records. Note added to advise that names of vulnerable components in the fragment section (FRAG) of the DATFIL are the same in the redundant section (RDNT) of the DATFIL.

Added target roll. The F/A-18 program office at China Lake requested that target roll be added to Velocity Triangle (MODENC=4), MODENC5 (MODENC=5), and Range Encounters (MODENC=6). Code was modified in JSEM subroutines ENCADD, FZENCN, RNGENC, and ZGENCN.

Demonstrated ability to process 62 views using PKTAB file. Normally PKTAB employs only 26 views; JSEM can process 62 views (like NMT methodology.)

Random Number Seed. An option for the user to select a random number seed based on system clock settings or the usual iseed.

APPENDIX C

JSEM 2.2 Errata Sheet.

The problems listed below will be corrected in JSEM 2.2.2. Included with the problem description is an interim workaround.

1. PROBLEM: Some error messages are written to the SUMPRT rather than all of them appearing in the DTLPRT. INTERIM WORKAROUND: None. This will be corrected in Version 2.2.2.

2. PROBLEM: GRFIL does not open when no file identification number exists (other files do open.) INTERIM WORKAROUND: Be sure to include GRFIL in the Unit5 file.

3. PROBLEM: If ENCFIL and DATFIL do not agree on the number of encounters, there is an abnormal termination of JSEM. INTERIM WORKAROUND: Be sure that ENCFIL and DATFIL have the same number of encounters in the data.

4. PROBLEM: Plotting of missile is incorrect if it intersects the contact model and is preemptively fuzed. INTERIM WORKAROUND: None. Plotting data will be off, but printed output is correct. This will be corrected in Version 2.2.2.

5. PROBLEM: If only one vulnerable component exists in the FRAG section of the DATFIL, the PKTAB file header will not be read. INTERIM WORKAROUND: Include more than one vulnerable component in the FRAG section of the DATFIL.

6. PROBLEM: The word PARAMETER is written twice when generating a PARAMETER statement using JPREP. This second appearance of the word PARAMETER causes the SGI and HP platforms to terminate a JSEM run with an error condition. INTERIM WORKAROUND: None. This will be corrected in Version 2.2.2.

7. PROBLEM: Total PK can be greater than one if KLLCNT = 1 is chosen, because Direct-Hit , Blast and Fragment are included. INTERIM WORKAROUND: None. This will be corrected in Version 2.2.2 .

8. PROBLEM: When running JSEM with KLLCNT = 1, JSEM does not write component information to the AVPKCP file if a blast kill occurs. INTERIM WORKAROUND: None. This will be corrected in Version 2.2.2.

9. PROBLEM: The message in DTLPRT file " POLCHK, POLYGON 1 IS CONCAVE. POLYGON MUST BE CONVEX AS THE COUNT OF THE NUMBER OF DIRECT HITS MAY BE AFFECTED. CHECK INPUT DATA." is sometimes caused by the program instead of the input file. INTERIM WORKAROUND: In subroutine POLCHK replace:

"IF (TOTANG .GE. (NCORNR-2)*PI) THEN"

with "IF (TOTANG .GE. (NCORNR-2)*3.14) THEN"

10. PROBLEM: For UNIX users, when creating a PARAM file in JPREP the program will terminate if a PARAM file already exists in the working directory. This is because it tries to open the PARAM file as STATUS=NEW INTERIM WORKAROUND: Be sure to remove any previous PARAM file from working directory when creating a new PARAM file with JPREP.

11. PROBLEM: When running multiple Monte Carlo encounters, the PK average may end up being greater than one. INTERIM WORKAROUND: Run only one Monte Carlo set of encounters at a time and manually average the PK.

12. PROBLEM: An error will occur if the velocity in the PKTAB file is not an integer. INTERIM WORKAROUND: Ensure velocities listed in the PKTAB file are integers.

13. PROBLEM: If STDOFF variable is not included in DATFIL, then no scaling for the EBC will take place regardless of setting of ISCEBC. INTERIM WORKAROUND: Include the STDOFF value following SCDIST in the DATFIL if scaling of the blast cylinder is desired.

14. PROBLEM: If the STPT fuzing mode is chosen, a mistake in the calculation of the fragment starting point for STPT fuzing mode may occur. INTERIM WORKAROUND: This only takes place when the STPT option is chosen. However, the results are not always incorrect.